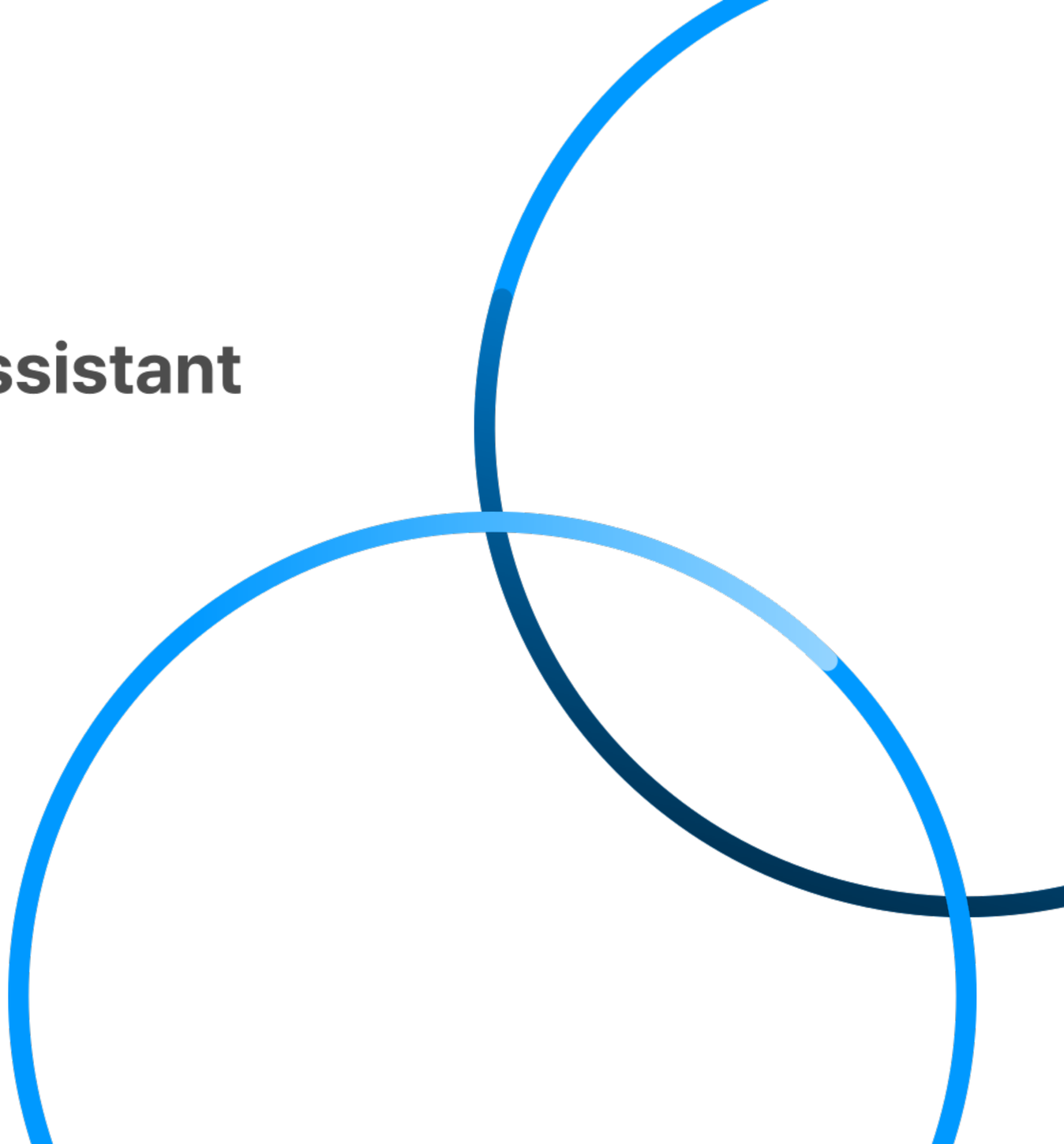
Capstone Design Project

"Mail AI" : AI-based Mail Assistant

The 3rd Bi-weekly Meeting

Team E: Yosep Kim, Soomin Lim, Dohyun Bu, Miseo Jeong



Agenda

Chapter 01

Introduction
- Timeline
- Teamwork

Chapter 02

Midterm Review
- Email Generation
- User Authentications
  - Login
  - Signup

Chapter 03

New Features
- Keyword Extration
  - each email
  - all email
- Email Summary
- CI/CD

Chapter 04

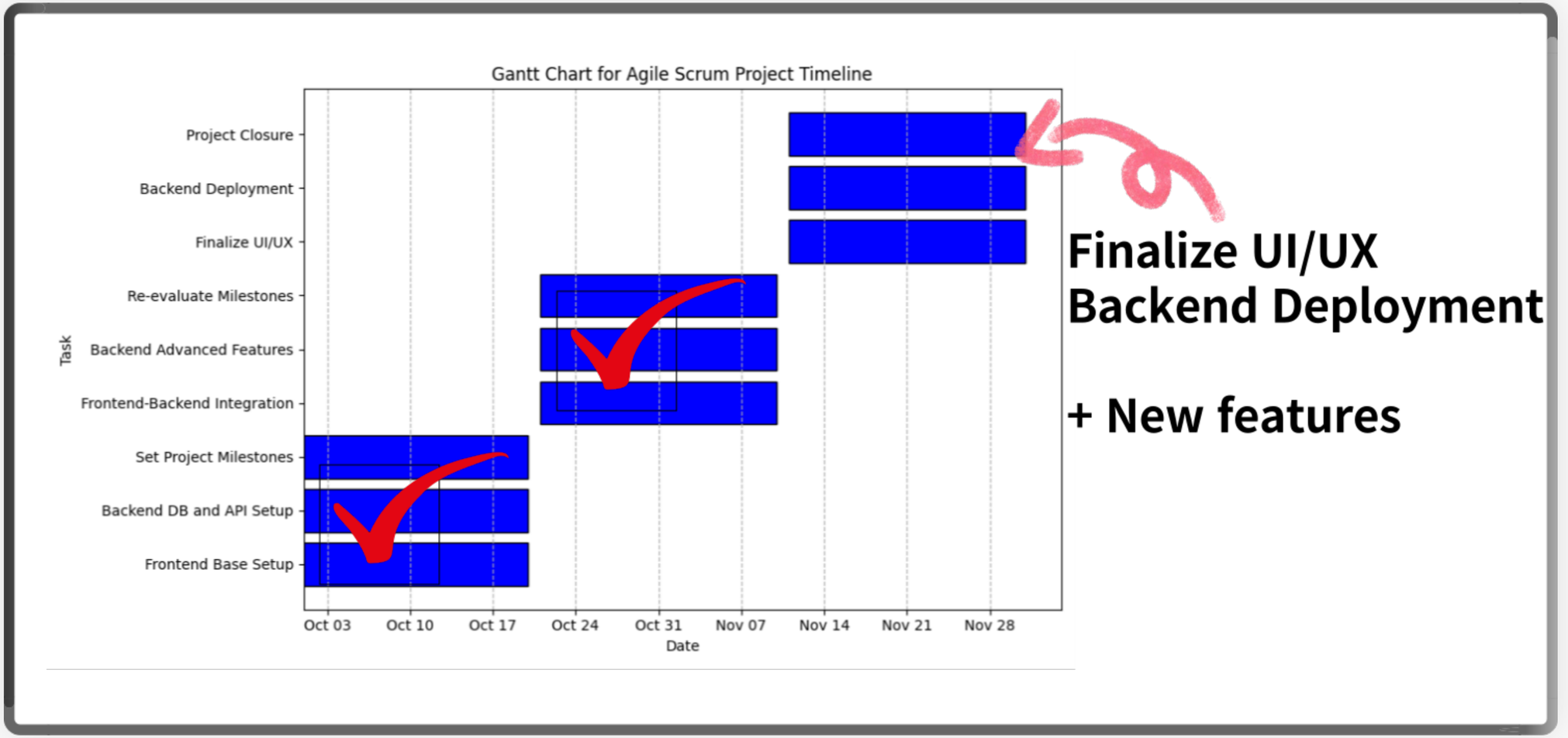
Conclusion
- Further Plans
- Q & A

Chapter
01

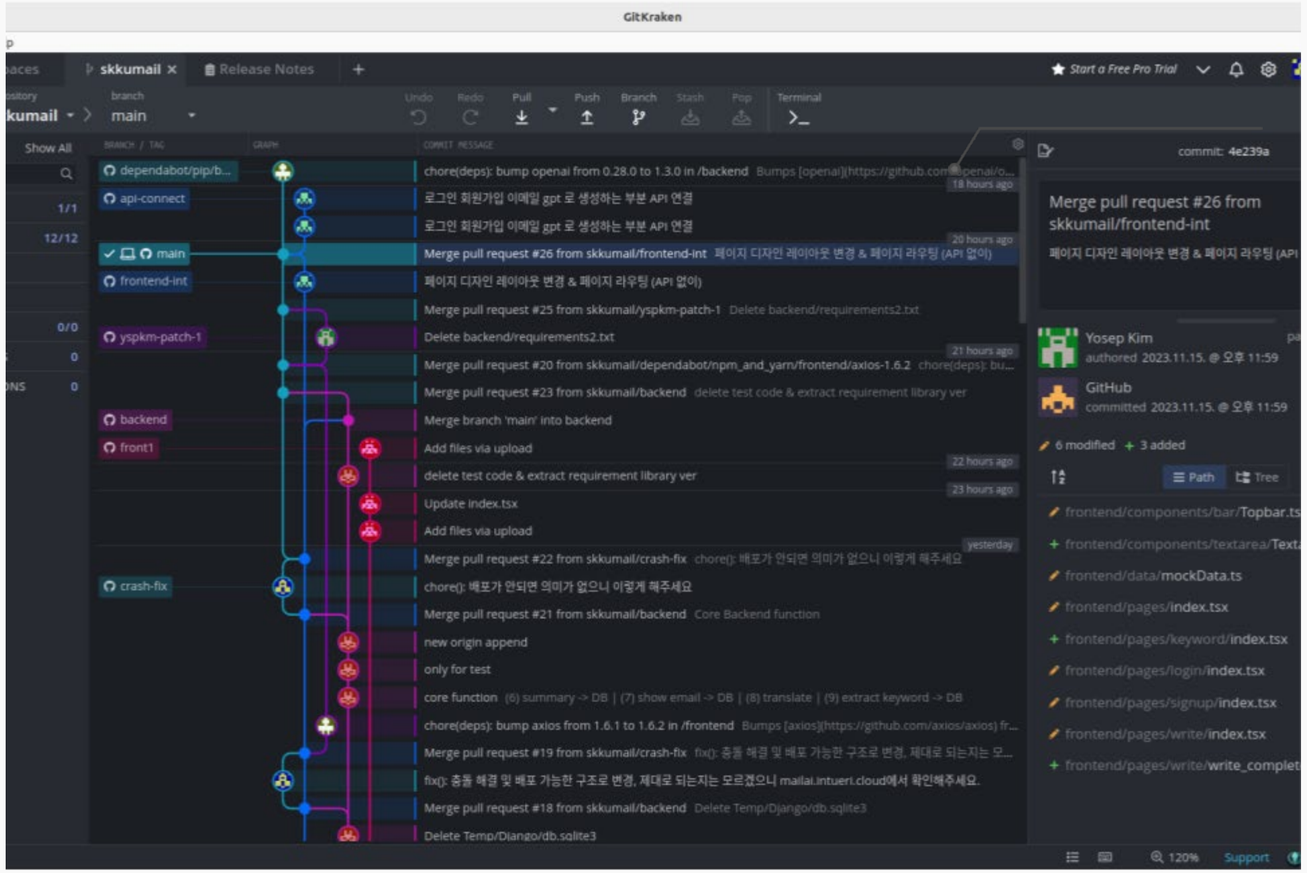
Introdction

- Timeline
- Team Role and Cooperations

Timeline: We are in 3rd Scrum!



Teamwork: Cooperation with GitHub and Git!



Team Roles

Team Member	Role & Responsibilities
Yosep Kim	Project Manager, Backend Development
Sumin Lim	Frontend Development (Next.js, UX/UI)
Dohyeon Boo	Backend Development (FastAPI, GPT, DB)
Miseo Jeong	Frontend Development (Next.js, UX/UI)

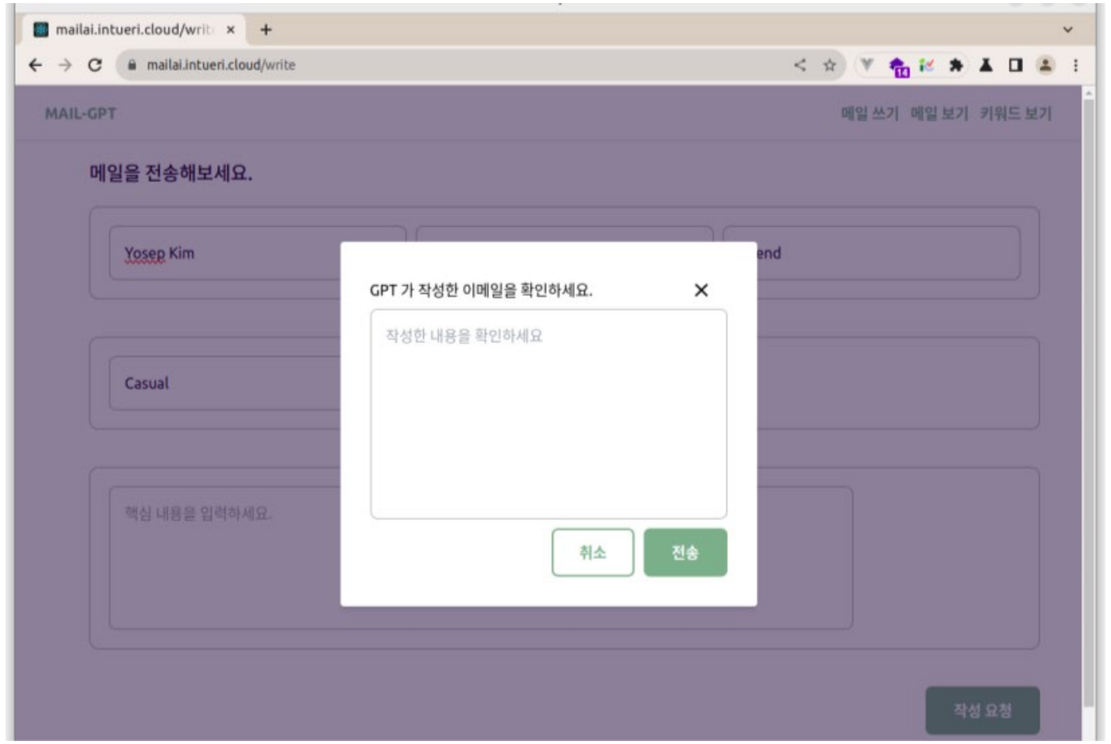
Commits	Branches	Merges
93	12	32

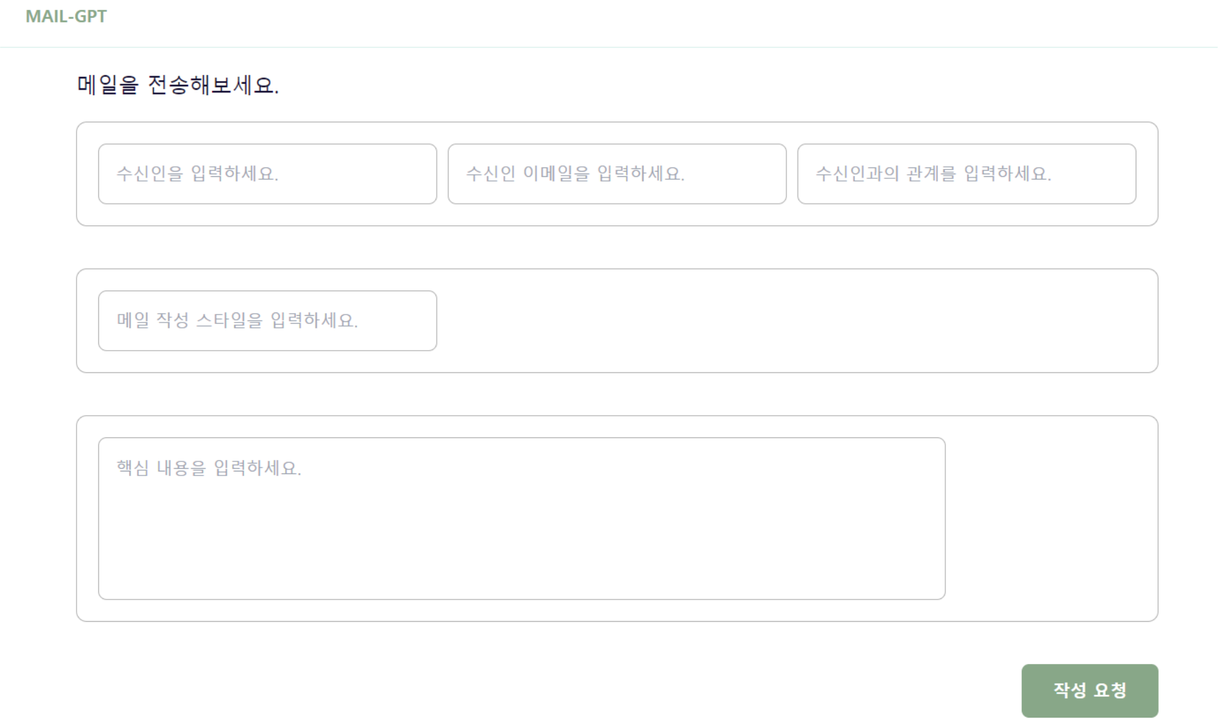
Chapter
02

Midterm Review

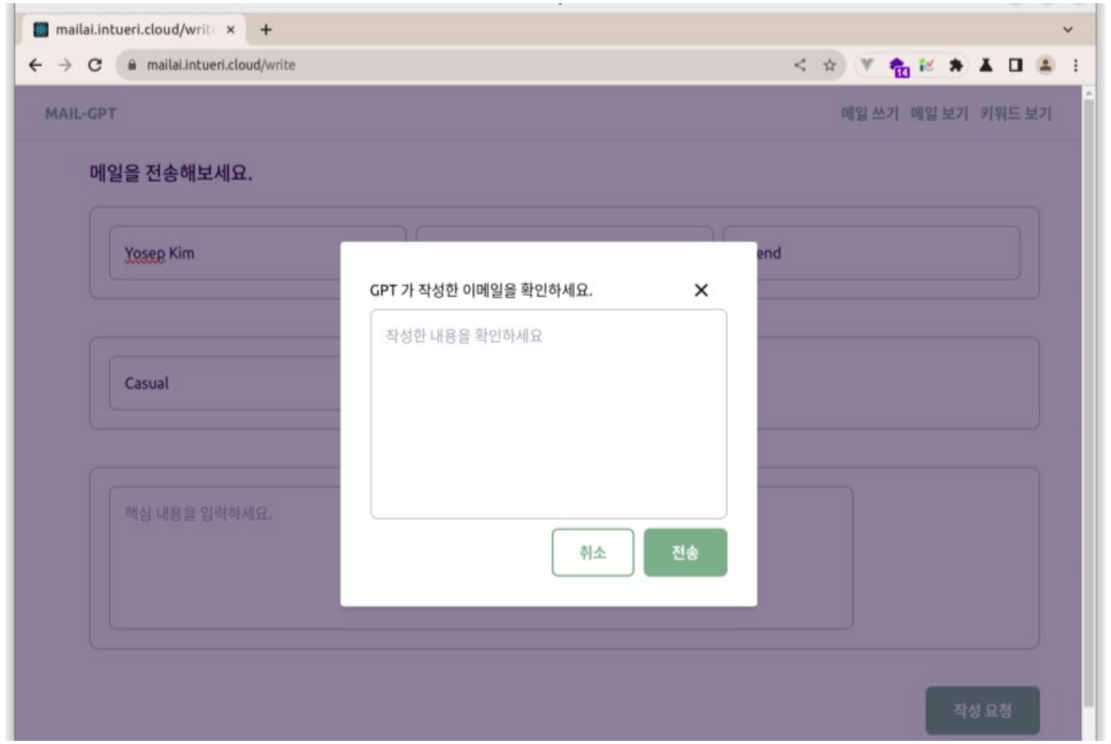
- Email Generation
- User Authentications
  - Login
  - Signup

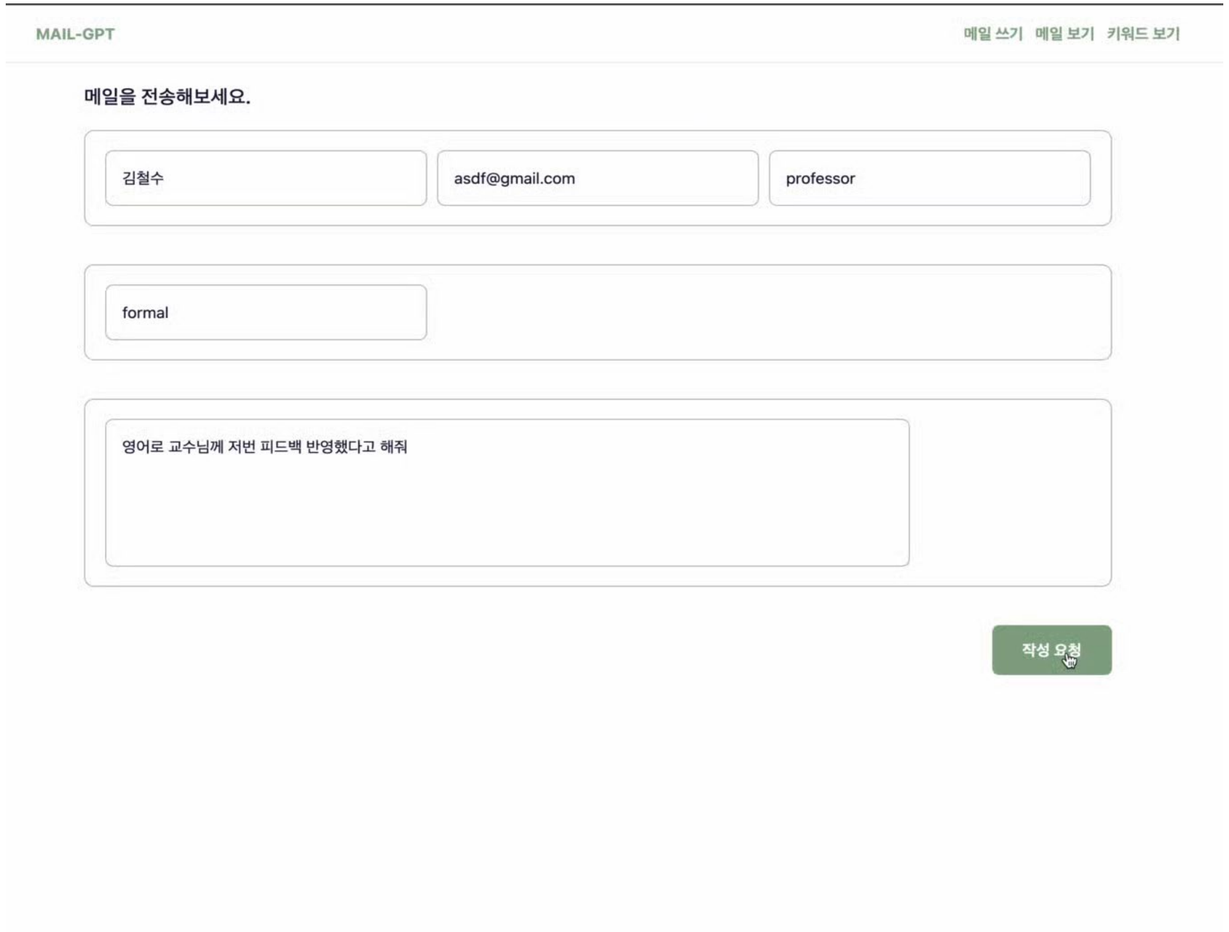
Email Generation: Frontend & Demo



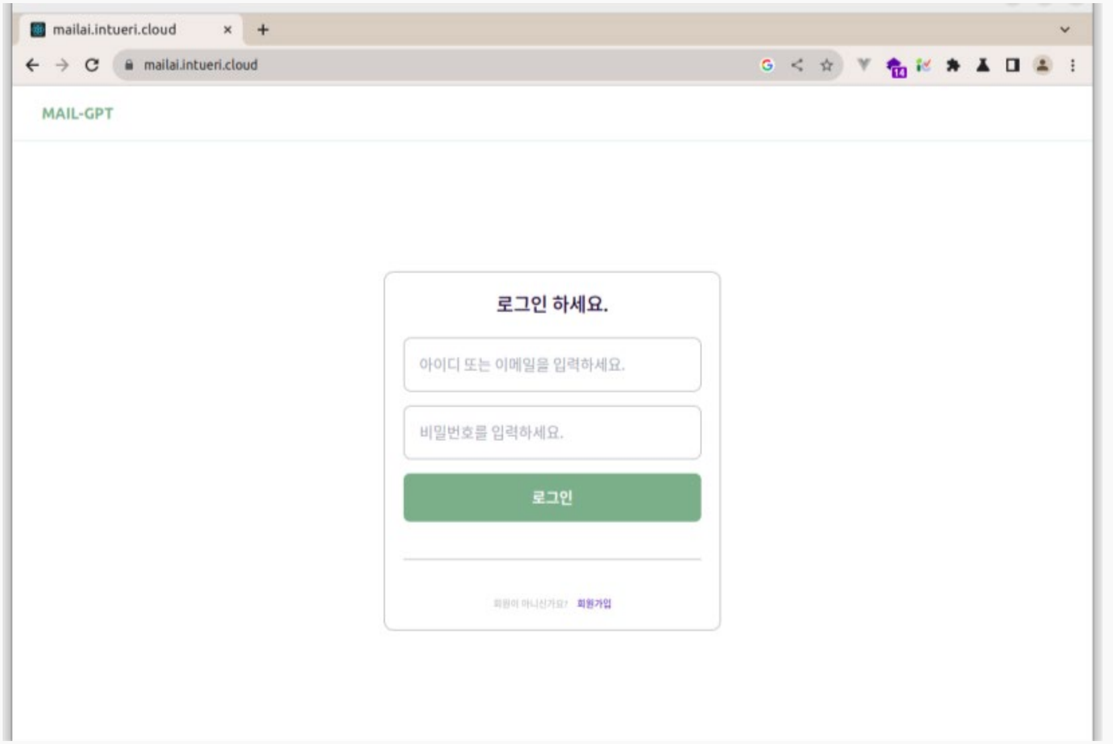


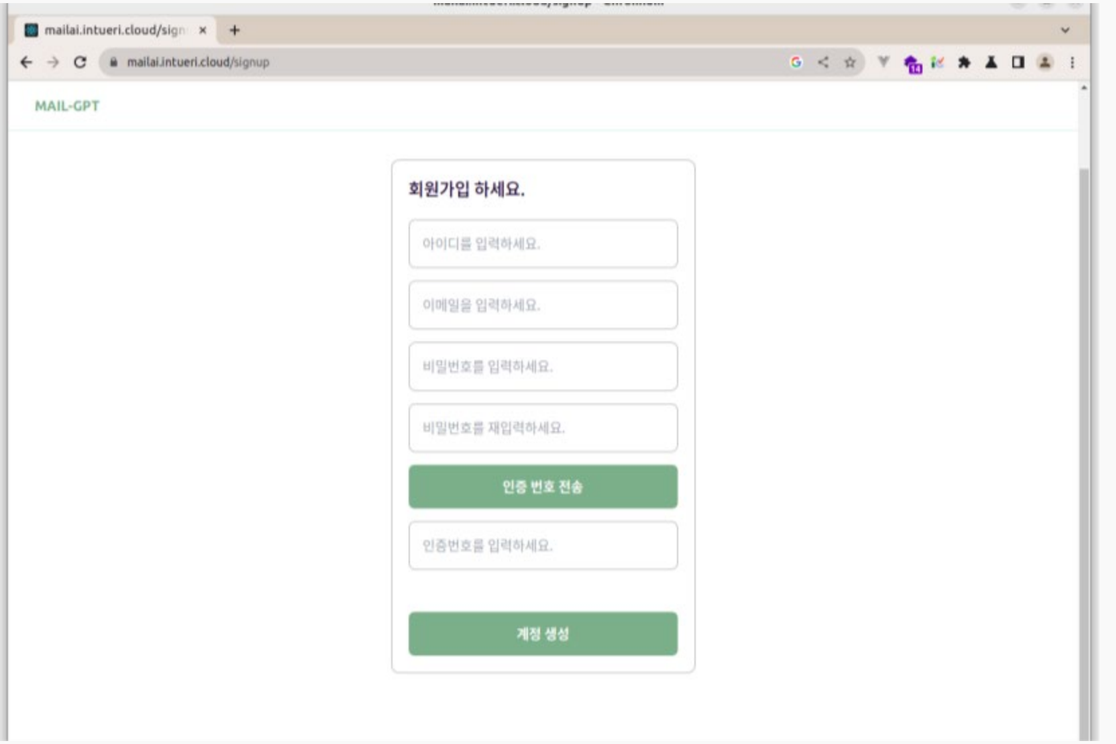
Email Generation: Frontend & Demo





User Authenticaiton: Login & Signup



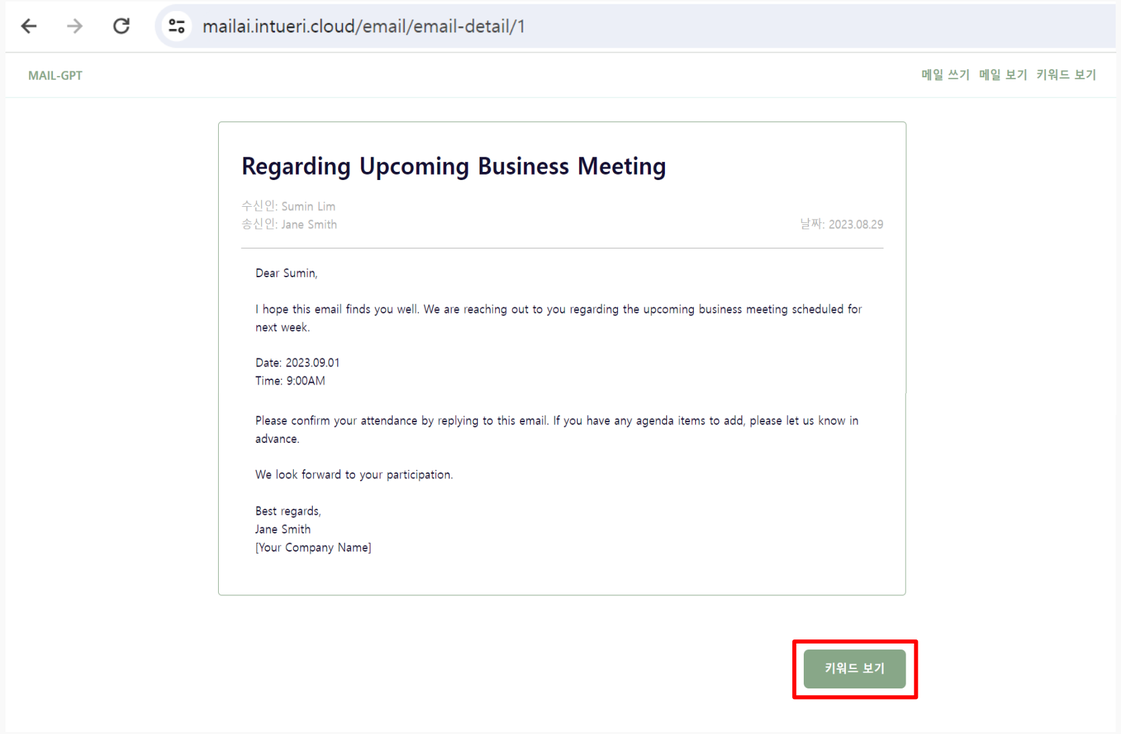


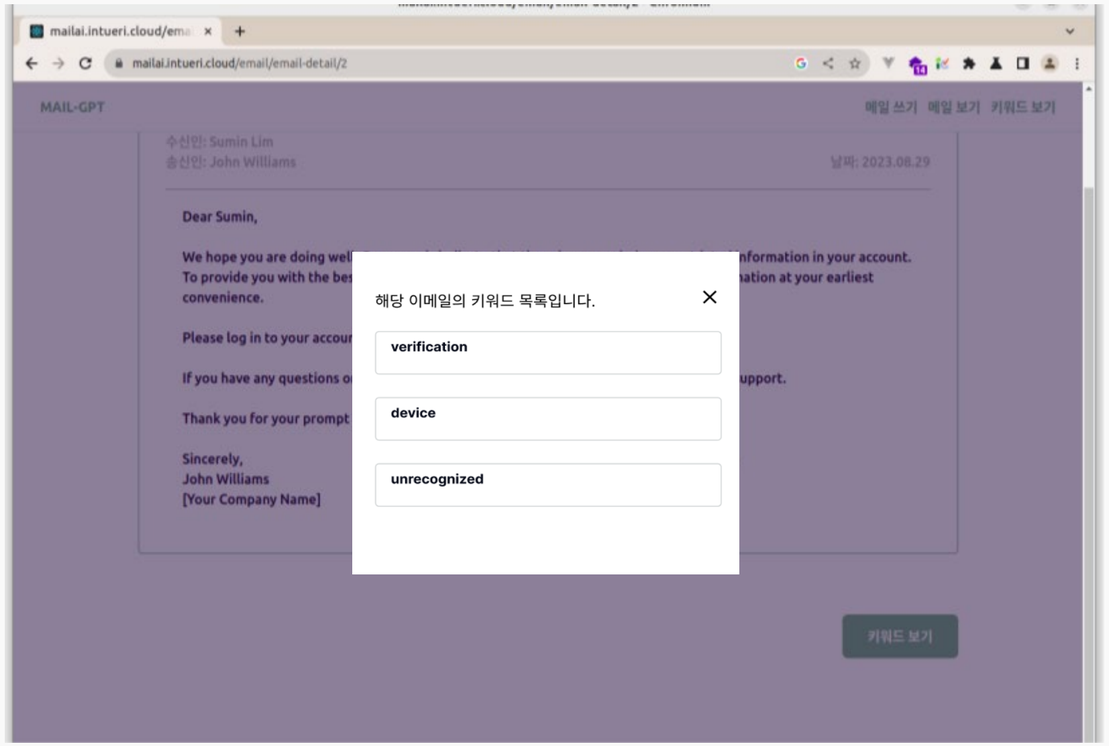
Chapter
03

New Features

- Keyword Extration
  - each email
  - all email
- Email Translation
- CI/CD

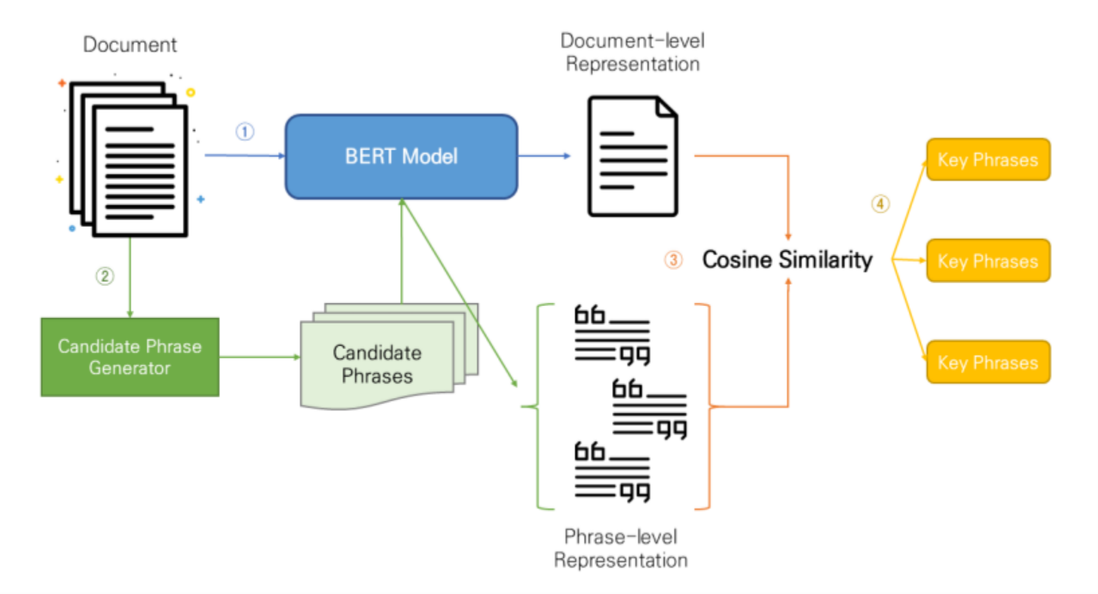
Keyword Extraction: Each & All





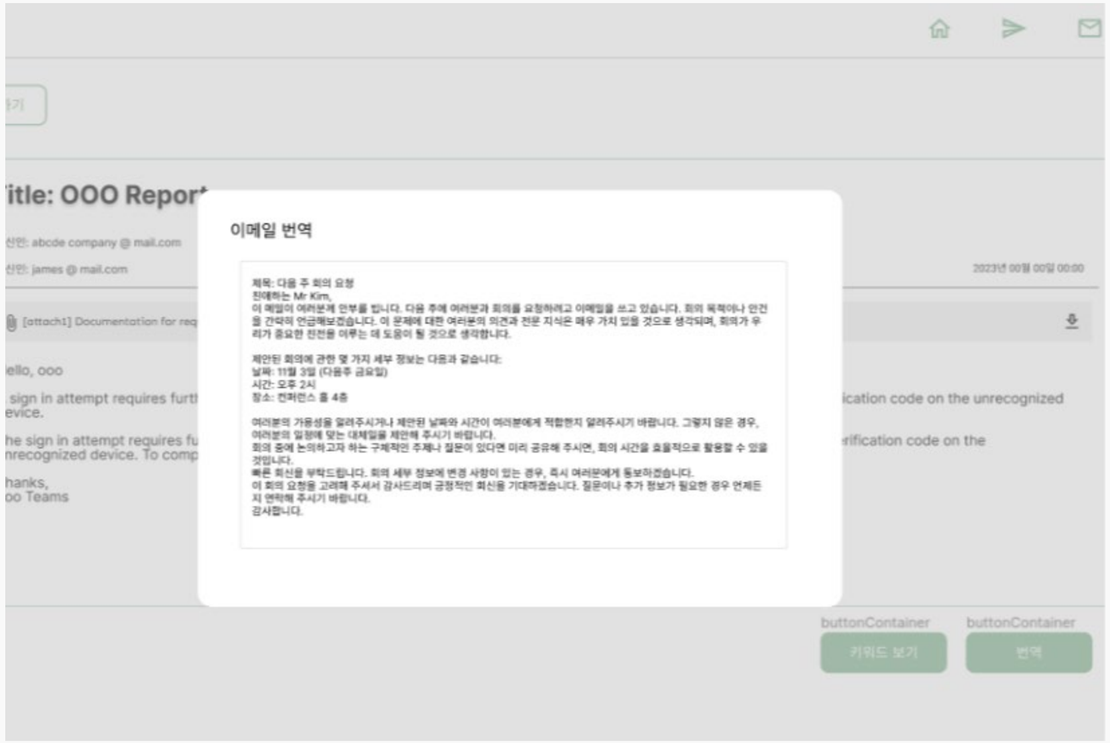
Keyword Extraction: Architecture

Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labeled training data consisting of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in a 'reasonable' way (see inductive bias).




```
        )
        # KeyBERT를 사용하여 키워드 추출
        keywords = keybert_model.extract_keywords(
            email_text, keyphrase_ngram_range=(1, 1), top_n = 5, stop_words="engl
        )

        total_list = []
        for i in range(len(keywords)):
            total_list.append(keywords[i][0])

        temp_user_msg.keyword = str(total_list)
        db.add(temp_user_msg)
        db.commit()

    # 추출된 키워드를 JSON 형식으로 반환
    return JSONResponse({'message': '연결 성공', "data":total_list})
except Exception as e:
    # 오류가 발생한 경우 500 Internal Server Error 반환
    raise HTTPException(status_code=500, detail=f"An error occurred: {str(e)}")
```

Email Translation: Google Translator API




```
################################ Translate ####################
@app.post("/translate")
async def translate_text(request: Request, db: Session=Depends(ge
    try:
        data = await request.json()

        pre_text = data.get('contents')

        translator = Translator()
        result = translator.translate(pre_text, dest='ko').text


        return JSONResponse({'message': '연결 성공', 'data':result}

    except Exception as e:
        return JSONResponse({'message': '연결 fail', 'data':str(e)
```

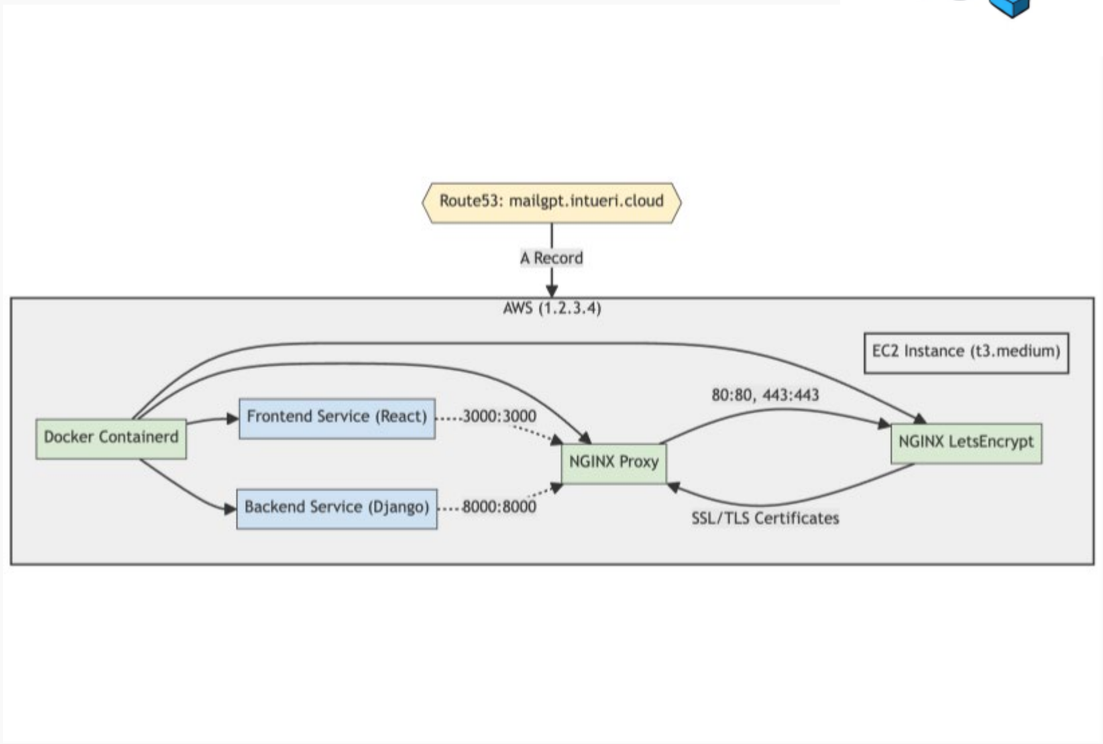
CI/CD Docker

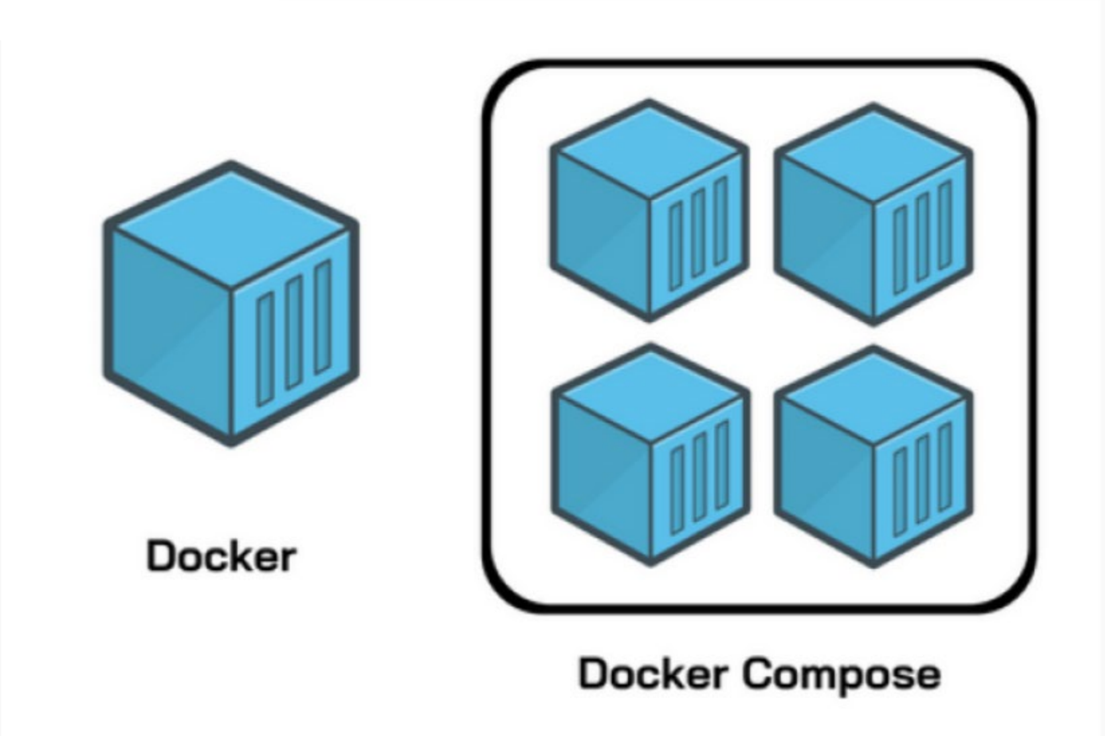

```dockerfile
FROM rockylinux/rockylinux:9.2

RUN dnf -y update && \
    dnf -y install gcc make python3 python3-pip python3-deve
    dnf clean all

WORKDIR /app

COPY requirements.txt .

RUN pip3 install --no-cache-dir -r requirements.txt

COPY . .

CMD ["uvicorn", "main:app", "--host", "0.0.0.0", "--port", "
```

```dockerfile
FROM ubuntu:jammy-20231004

RUN apt-get update \
       && apt-get upgrade -y \
       && apt-get install -y ca-certificates curl gnupg \
       && mkdir -p /etc/apt/keyrings \
       && curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key \
              | gpg --dearmor -o /etc/apt/keyrings/nodesource.gpg

ENV NODE_MAJOR=20

RUN echo "deb [signed-by=/etc/apt/keyrings/nodesource.gpg] https://deb.nodesource.com/node_$NODE_MAJOR.x nodistr
       | tee /etc/apt/sources.list.d/nodesource.list \
       && apt-get update \
       && apt-get install nodejs -y

WORKDIR /app
COPY package.json .

RUN npm install

COPY . .
RUN npm run build

EXPOSE 3000

CMD ["npm", "start"]
```

FastAPI Rocky Linux™ NEXT.JS Ubuntu

CI/CD Docker Compose





Chapter
04

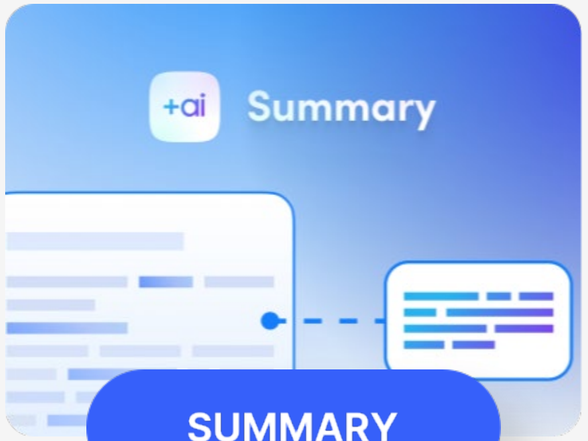
Conclusion

- Further Plans
- Q & A

Further Plan

EMAIL DB

CLUSTERING



SUMMARY


Q&A

Capstone Design Project

THANK YOU

FOR WATCHING!